

<LEI\_ESII2021\_TP2\_GRUPO5>

TEST DESIGN SPECIFICATION

CLASS MARKETPLACELDGER

---

Version <1.2>

<19/01/2022>

### Histórico de Versões

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Rui Neto	11-01-2022	Simão Santos	11-01-2022	Início da doc.
1.1	Rui Neto	16-01-2022	Simão Santos	16-01-2022	Continuação da doc.
1.2	Rui Neto	19-01-2022	Simão Santos	19-01-2022	Resolução de ID duplicado.

## Índice

1. Introdução .....	4
1.1. Identificador do documento .....	4
1.2. Âmbito .....	4
1.3. Referências .....	4
2. Features/Itens a testar .....	5
3. Detalhes da abordagem aos testes .....	6
Método addOrderRequest() .....	6
Método registerOrderInLedger() .....	6
Método getOrderRequest() .....	7
Método removeOrderRequest() .....	7
Método getOrdersCounter() .....	8
4. Identificação dos Testes .....	10
5. Critérios de passagem ou falha das features .....	11

# 1. Introdução

## 1.1. Identificador do documento

[TCS-10]MarketPlaceLedger

## 1.2. Âmbito

O documento foca-se na testegam das funcionalidades da Class MarketPlaceLedger.

Items no âmbito: Testes funcionais da class MarketPlaceLedger.

Items fora do âmbito: Testes à Base de dados, hardware, alguma interface externa e performance.

## 1.3. Referências

[TCS-1]BasicOrganization

[TCS-3]BasicBlock

[TCS-6]MarketPlaceEntity

[TCS-8]ProductsTransaction

[TCS-9]PaymentTransaction

## 2. Features/Itens a testar

Item a testar	Descrição	UserStory	Responsabilidade
addOrderRequest()	Adiciona uma productTransaction à lista de OrderRequests.	#2	Alta
registerOrderInLedger()	Regista no Ledger as orders pagas. Recebe como parâmetro um array de paymentTransactions.	#2	Alta
getOrderRequest()	Devolve a productTransaction com o ID passado como parâmetro, que esteja na lista de OrderRequests.	#2	Alta
removeOrderRequest()	Remove a productTransaction com o ID passado como parâmetro, que esteja na lista de OrderRequests.	#2	Alta
getOrdersCounter()	Devolve o número de Orders registadas no Ledger.	#3	Alta
averageTransactionsValue() ( )	Calcula o valor médio das transações.	#5	Média
averageNumberOfProductsPerTransaction() sPerTransaction()	Retorna a média do número de produtos por transação.	#7	Baixa
averageNumberOfSalesPerDistrict() rDistrict()	Retorna a média do valor de vendas por distrito.	#6	Baixa
averageNumberOfPurchasesPerDistrict() esPerDistrict()	Retorna a média do valor de compras por distrito.	#6	Baixa

### 3. Detalhes da abordagem aos testes

#### Método addOrderRequest()

##### Pré-requisitos

- Existir uma instância da classe MarketPlaceLedger
- O input é do tipo ArrayList<ProductsTransaction>

##### Classes de equivalência

EC1 -> O output é do tipo int

EC2 -> O output não é do tipo int

EC3 -> Retorna corretamente o número de transações adicionadas

EC4 -> Não retorna corretamente o número de transações adicionadas

EC5 -> Não permite adicionar orders com o mesmo ID

EC6 -> Permite adicionar orders com o mesmo ID

Condição	Classe de Eq. válida	Classe de Eq. inválida
Tipo do output	EC1	EC2
Valor retornado	EC3	EC4
Adição de orders	EC5	EC6

#### Método registerOrderInLedger()

##### Pré-requisitos

- Existir uma instância da classe MarketPlaceLedger
- O input é do tipo ArrayList<PaymentTransaction>

##### Classes de equivalência

EC1 -> As orders registadas saem das orders requests

EC2 -> As orders registadas não saem das orders requests

EC3 -> Tanto a productTransaction como a paymentTransaction são adicionadas ao Ledger e estão validadas

EC4 -> Nem a productTransaction nem a paymentTransaction são adicionadas ao Ledger e estão validadas

EC5 -> O output é do tipo 'Iterator<ProductsTransaction>'

EC6 -> O output não é do tipo 'Iterator<ProductsTransaction>'

EC7 -> Retorna o iterador apenas com as product transactions pagas

EC8 -> Não retorna o iterador apenas com as orders pagas

Condição	Classe de Eq. válida	Classe de Eq. inválida
Orders registadas	EC1	EC2
Transações adicionadas ao Ledger	EC3	EC4
Tipo do output	EC5	EC6
Valor output	EC7	EC8

## Método getOrderRequest()

### Pré-requisitos

- O input é do tipo 'String'.

### Classes de equivalência

EC1 -> O output é do tipo 'ProductstTransaction'

EC2 -> O output não é do tipo 'ProductstTransaction'

EC3 -> Retorna a order request corretamente

EC4 -> Não retorna a order request corretamente

Condição	Classe de Eq. válida	Classe de Eq. inválida
Tipo do output	EC1	EC2
Valor retornado	EC3	EC4

## Método removeOrderRequest()

### Pré-requisitos

- O input é do tipo 'String'.

### Classes de equivalência

EC1 -> O output é do tipo 'boolean'

EC2 -> O output não é do tipo 'boolean'

EC3 -> Retorna true e a order request é removida

EC4 -> Retorna true e a order request não é removida

EC5 -> Retorna false e order request não é removida

EC6 -> Retorna false e a order request é removida

Condição	Classe de Eq. válida	Classe de Eq. inválida
Tipo do output	EC1	EC2
Valor retornado	EC3, EC5	EC4, EC6

## Método getOrdersCounter()

### Pré-requisitos

- Existir uma instância da classe 'MarketPlaceLedger'

### Classes de equivalência

EC1 -> O output é do tipo 'int'

EC2 -> O output não é do tipo 'int'

EC3 -> Retorna corretamente o número de Orders no Ledger

EC4 -> Não retorna corretamente o número de Orders no Ledger

Condição	Classe de Eq. válida	Classe de Eq. inválida
Tipo do output	EC1	EC2
Valor retornado	EC3	EC4

## Método averageTransactionsValue()

### Pré-requisitos

- Existir uma instância da classe 'MarketPlaceLedger'

### Classes de equivalência

EC1 -> O output é do tipo 'float'

EC2 -> O output não é do tipo 'float'

EC3 -> Retorna corretamente a média do valor das transações do Ledger

EC4 -> Não retorna corretamente a média do valor das transações do Ledger

Condição	Classe de Eq. válida	Classe de Eq. inválida
Tipo do output	EC1	EC2
Valor retornado	EC3	EC4

## Método averageNumberOfProductsPerTransaction()

### Pré-requisitos

- Existir uma instância da classe 'MarketPlaceLedger'

### Classes de equivalência

EC1 -> O output é do tipo 'float'

EC2 -> O output não é do tipo 'float'

EC3 -> Retorna corretamente a média do número de Produtos por transação

EC4 -> Não retorna corretamente a média do número de Produtos por transação

Condição	Classe de Eq. válida	Classe de Eq. inválida
Tipo do output	EC1	EC2
Valor retornado	EC3	EC4



## Método averageValueOfSalesPerDistrict()

### Pré-requisitos

- Existir uma instância da classe 'MarketPlaceLedger'

### Classes de equivalência

EC1 -> O output é do tipo 'float'

EC2 -> O output não é do tipo 'float'

EC3 -> Retorna corretamente a média do valor de vendas por Distrito

EC4 -> Não retorna corretamente a média do valor de vendas por Distrito

Condição	Classe de Eq. válida	Classe de Eq. inválida
Tipo do output	EC1	EC2
Valor retornado	EC3	EC4

## Método averageValueOfPurchasesPerDistrict()

### Pré-requisitos

- Existir uma instância da classe 'MarketPlaceLedger'

### Classes de equivalência

EC1 -> O output é do tipo 'float'

EC2 -> O output não é do tipo 'float'

EC3 -> Retorna corretamente a média do valor de compras por Distrito

EC4 -> Não retorna corretamente a média do valor de compras por Distrito

Condição	Classe de Eq. válida	Classe de Eq. inválida
Tipo do output	EC1	EC2
Valor retornado	EC3	EC4

## 4. Identificação dos Testes

TestCase	Método	Descrição
TC_MPL_1	addOrderRequest()	Verificar resultados ao adicionar uma ArrayList de productsTransaction vazia.
TC_MPL_2	addOrderRequest()	Verificar resultados ao adicionar uma ArrayList de productsTransaction com uma productsTransaction.
TC_MPL_3	addOrderRequest()	Verificar resultados ao adicionar uma ArrayList de productsTransaction com duas productsTransaction.
TC_MPL_4	registerOrderInLedger()	Verificar se Orders registadas saem da lista de order requests.
TC_MPL_5	registerOrderInLedger()	Verificar se tanto as paymentTransactions como as productsTransactions não são adicionadas ao Ledger se valor não for coberto.
TC_MPL_6	registerOrderInLedger()	Verificar resultados ao registar 0 orders.
TC_MPL_7	getOrderRequest()	Verificar resultados ao passar como parâmetro um id de uma order request que não existe.
TC_MPL_8	getOrderRequest()	Verificar resultados ao passar como parâmetro um id de uma order request que existe.
TC_MPL_9	removeOrderRequest()	Verificar resultados ao passar como parâmetro um id de uma order request que não existe.
TC_MPL_10	removeOrderRequest()	Verificar resultados ao passar como parâmetro um id de uma order request já removida.
TC_MPL_11	getOrdersCounter()	Verificar resultados com 0 Orders no Ledger.
TC_MPL_12	getOrdersCounter()	Verificar resultados com 1 Order no Ledger.
TC_MPL_13	averageTransactionsValue()	Verificar resultados ao invocar método com o Ledger vazio.
TC_MPL_14	averageTransactionsValue()	Verificar resultados com duas orders no Ledger.
TC_MPL_15	averageNumberOfProductsPerTransaction()	Verificar resultados ao invocar método com 0 orders no Ledger.
TC_MPL_16	averageNumberOfProductsPerTransaction()	Verificar resultados ao invocar método com 2 orders no Ledger.
TC_MPL_17	averageNumberOfSalesPerDistrict()	Verificar resultados ao invocar método com 0 orders no Ledger.
TC_MPL_18	averageNumberOfSalesPerDistrict()	Verificar resultados ao invocar método com 2 orders no Ledger.
TC_MPL_19	averageNumberOfPurchasesPerDistrict()	Verificar resultados ao invocar método com 0 orders no Ledger.
TC_MPL_20	averageNumberOfPurchasesPerDistrict()	Verificar resultados ao invocar método com 2 orders no Ledger.
TC_MPL_21	addOrderRequest()	Verificar resultados ao adicionar uma ArrayList de productsTransaction com duas transações com o mesmo ID.

## 5. Critérios de passagem ou falha das features

Features	Critérios de passagem
Registrar Orders no Ledger	Regista no Ledger apenas as orders pagas.
Retornar corretamente a informação do Ledger e das orders registadas no Ledger.	Retorna corretamente o número de Orders no Ledger. Retorna corretamente as ordres do Ledger.